Kapitel 2 – Kontrollstrukturen: Zusammenfassung + Übungen

In diesem Kapitel lernst du, wie dein Programm Entscheidungen trifft: mit Bedingungen (`if`, `else`, `switch`) und wie man Anweisungen logisch gruppiert. Kontrollstrukturen sind entscheidend für die Steuerung des Programmablaufs.

Themenübersicht

1. Blöcke mit { }

Mehrere Anweisungen zu einem logischen Block gruppieren. Wichtig für Bedingungen.

2. if-Anweisung

Führt Anweisungen aus, wenn eine Bedingung wahr ist:
if (x > 5) { ... }

3. if-else

Wähle zwischen zwei Alternativen:
if (x > 5) { ... } else { ... }

4. else if – mehrere Fälle

Mehrere Alternativen nacheinander prüfen:
if (...) {...} else if (...) {...} else {...}

5. Das Dangling-else-Problem

Unklare Zuordnung bei verschachtelten if-Anweisungen. Immer Klammern setzen!

6. switch-Anweisung

Alternative zu vielen if-else: switch(variable) { case ...: break; ... }

7. default im switch

Wird ausgeführt, wenn kein anderer Fall zutrifft.

8. break im switch

Beendet die aktuelle case-Ausführung. Ohne break wird weiter ausgeführt (fall-through).

Übungsaufgaben

1. 1. Schreibe ein Programm, das prüft, ob eine eingegebene Zahl positiv, negativ oder 0 ist.

2. 2. Verwende if-else, um das Alter zu prüfen:
   Unter 18: 'zu jung', 18-65: 'erwachsen', darüber: 'Rentner'.
3. 3. Implementiere eine if-Anweisung mit einem Block aus 3 Anweisungen.
4. 4. Erkläre mit Beispiel das Dangling-else-Problem.
5. 5. Nutze switch, um bei den Zahlen 1–3 den passenden Wochentag (Mo–Mi) auszugeben.
6. 6. Was passiert, wenn man im switch kein break verwendet? Erkläre mit Beispiel.
7. 7. Verwende default im switch, um alle anderen Zahlen als 'ungültig' zu behandeln.
8. 8. Baue eine verschachtelte if-Abfrage mit Klammern zur besseren Lesbarkeit.